

OASIS: A Graphical Decision Support System for Ground-Water Contaminant Modeling

by Charles J. Newell^a, John F. Haasbeek^b, and Philip B. Bedient^b

Abstract

Three new software technologies were applied to develop an efficient and easy to use decision support system for ground-water contaminant modeling. Graphical interfaces create a more intuitive and effective form of communication with the computer compared to text-based interfaces. Concepts from the field of hypertext were applied to design the extensive electronic documentation and databases. Finally, object oriented programming permitted scientists with little programming experience to develop the system by manipulating preexisting software objects instead of writing computer code, greatly increasing the productivity of the project team.

A decision support system is a class of software with different origins and characteristics than conventional programs or expert systems. Decision support systems are designed to help users with broad, semistructured problems, with the user directing the problem-solving process. The decision support software described in this paper, the OASIS system, provides ground-water modelers with a ground-water biodegradation model, model documentation, background information, data from chemical and hydrogeologic databases, and data management tools. The system was developed using HyperCard software on a Macintosh personal computer. It currently has over 1700 different computer screens of information and occupies approximately 10 megabytes of hard disk storage.

OASIS is a new type of modeling software where ground-water models, data, and knowledge are integrated together using a graphical interface and an easily modified software architecture. Two groups of users can benefit from the system: current modelers who need more efficient interfaces and data management tools, and people who are not using models now because the modeling process is too involved and requires specialized knowledge.

Introduction

New software technologies are encouraging the development of new types of computer software systems with combinations of information, data, and computational tools. Three such technologies, graphical interfaces, hypertext, and object oriented programming, were incorporated into the design of OASIS, a new decision support system for ground-water contaminant modeling at hazardous waste sites. OASIS is intended for two groups of users: current modelers who need more efficient interfaces and data management systems, and people who are not using models now because the models are too complex and require specialized knowledge and computer skills.

The paper is divided into three parts. First, decision support systems are a general class of computer software with different origins and characteristics from expert systems and other types of software. Second, three new software technologies were applied in the design and development of the OASIS system. Finally, representative screens from the 1700 screens of documentation, databases,

data management tools, and interfaces that comprise OASIS illustrate a large, integrated software system.

Decision Support Systems

Origins of Decision Support Systems

The term "Decision Support System" (DSS) was first used in the business and management world to describe a new class of software for supporting executive decision-making instead of computing solutions to everyday operational problems. Gorry and Scott-Morton (1971) developed a framework for contrasting conventional software programs with decision support systems. Conventional software was characterized by the authors as automation for repetitive, highly structured problems, such as payroll and inventory where there is a well-defined procedure for processing information. A DSS, on the other hand, was defined as software for increasing the effectiveness of a decision-maker while he or she performs less structured tasks, typically managerial or planning in nature. These problems are nonroutine and nonrepetitive, and the decision-maker usually does not have an established approach to solving the problem. A DSS supports, but does not replace the decision-making process of the person solving the problem.

A further description of decision support systems was given by Sprague and Carlson (1982):

- They tend to be aimed at the less well-structured, underspecified problems that upper-level managers and professionals typically face.

^aGroundwater Services, Inc., Houston, Texas 77098.

^bNational Center for Ground Water Research, Department of Environmental Science and Engineering, Rice University, Houston, Texas 77251.

Received October 1988, revised May 1989, accepted June 1989.

Discussion open until September 1, 1990.

- They try to combine the use of models or analytical techniques with traditional data access and retrieval functions.
- They specifically focus on features that make them easy to use by noncomputer people in an interactive mode.
- They stress flexibility and adaptability to fit changes in the situation and decision-making approach of the user.

Decision Support Systems Architecture

Sprague (1987) views DSS as the intersection of two evolutionary trends from the field of data processing and the field of business modeling (management science). Data processing and databases have evolved from basic data processing to sophisticated databases with English-like query languages. Business modeling has also evolved from programs designed just to solve difficult mathematical problems to integrated data management, statistical, and report-generating systems. Decision support systems lie at the intersection of these two trends, where databases and models are integrated together in a common software environment via a common interface, so that the data and the models are easier to use together. The integration of data, models, and a sophisticated user interface is a key characteristic of a decision support system.

Development of Decision Support Systems

Conventional software is developed with a structured programming or life cycle approach. The development process is divided into planned stages, such as specification/design, coding, testing, installation, and maintenance, with specification/design requiring the largest share of the effort. In structured programming projects, coding is typically delayed until a complete design of the program has been laid out on paper.

Decision support systems are built using a different design philosophy. Keen (1980) describes the DSS development process as a "middle-out" design, where prototypes of the software are developed early in the project, prospective users evaluate the software, and a cycle of changes and evaluation is conducted. The emphasis is to start the actual software "coding" as soon as possible, and cycle through a series of improvements until the software evaluators show the software is useable.

Decision Support Systems vs. Expert Systems

Decision support systems are a new class of computer software being applied to semistructured problems difficult or impractical to solve using conventional structured programming methods, such as FORTRAN or C. Another class of software designed to deal with the same type of problems are expert systems. The OASIS project originally began as a pure application of expert systems technology that evolved into a decision support system approach. Because of the expert system history of OASIS, and the similar goals of decision support and expert systems software, these two types of decision-making software are compared below.

Expert systems originated from the field of artificial intelligence, where a general goal is to embody a computer

with human-like intelligence. Early approaches to artificial intelligence concentrated on the general process of reasoning. By the 1970's the limits to this approach led to the hypothesis that the knowledge a person has, rather than the patterns of reasoning he uses, is the key to mimicking human intelligence (Feigenbaum, 1977). This hypothesis is the essence of the expert systems approach. The programmer concentrates on representing the knowledge an expert uses to answer a specific question, instead of trying to computerize general reasoning principles that can be used for any problem. One common method used by expert systems developers to represent the knowledge are "If-Then" type rules developed from intensive interviews with an expert. Once the expert system has been developed, the computer "asks" the user questions and then formulates an answer to the problem.

An expert system is usually applied to narrow and very specific problems. The knowledge an expert has about a particular topic is enormous, and developers of expert systems must stay within a very small area to limit the amount of knowledge that is needed to answer the problem. Expert systems are usually system-driven; the computer dictates what the next step is, usually by asking the user a question. Because expert systems are expensive to develop, they usually deal with operational problems that need to be answered frequently by users (Doukidis, 1988). In summary, an expert system is an attempt to impose structure on a semistructured problem by trying to extract rules from an expert that have never been recorded or expressed.

Decision support systems, on the other hand, are more like a collection of tools and data that are used to solve problems. The problems are often very broad; the user can play with the tools and experiment to find a solution. The decision process is controlled by the user when he or she decides how to apply the different types of tools and data that are available. Generally, DSS are best suited to solving management and planning domain, instead of routine operational problems. The DSS are not computerized expert; they are a computerized assistant for the user.

Using a ground-water contaminant transport model is a semistructured problem that is amenable to the decision support systems approach. A model provides some structure to the problem, but the modeler must make a series of decisions about how to set up the model and represent the site, how to apply all the available data, and which simulations should be made. It would be very difficult to write a conventional program (in FORTRAN, for example) that could take a description of a problem, all the data that are available, and then run the model. An expert systems approach would also be difficult to apply because the scope of the problem is large; there are many different types of modeling studies, different scales of problems, and different types of data involved in using a model. It would be difficult to incorporate all of this information into an expert system that could "guide" a user through a general modeling process, unless the scope of the problem was limited to a narrow class of ground-water modeling problems. The OASIS project evolved from a narrow expert systems approach into a decision support systems approach

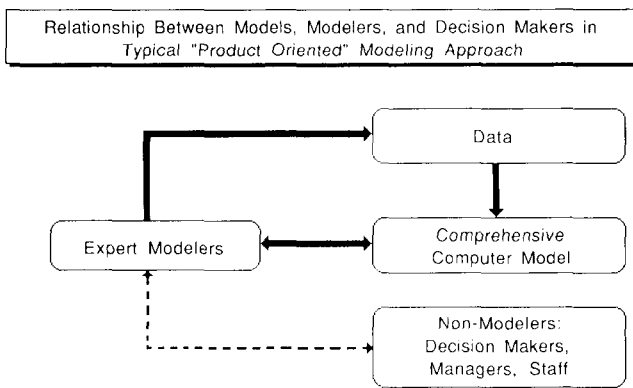


Fig. 1. Diagram of "product oriented" modeling.

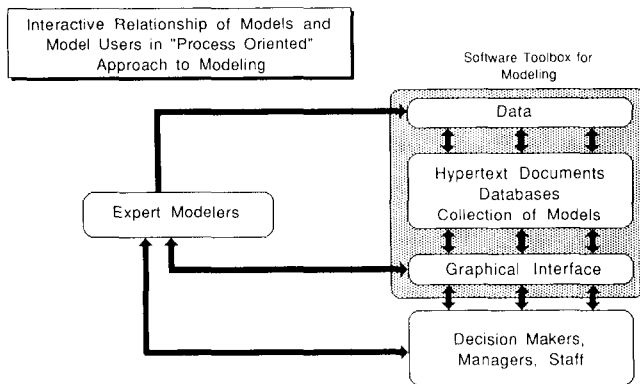


Fig. 2. Diagram of "process oriented" modeling.

providing assistance with more general problems. OASIS provides modelers with a collection of tools controlled by the modeler (not by the computer) to solve a ground-water problem.

Decision Support Systems in Environmental Engineering

Some authors have suggested that decision support systems can increase the utilization of environmental modeling and enhance technical communication. Loucks, Kindler, and Fedra (1985) introduced a series of five papers on the state of current modeling practices and discuss the benefits of interactive decision support software for water resources. They evaluated the constraints to the widespread use of environmental modeling and suggested that technical problems, such as poor scientific representations of natural processes, were not as significant as institutional constraints, such as difficulties people have in interacting with the model itself.

They suggested a shift from the existing "product" oriented approach to a "process" oriented approach. The product oriented approach occurs when modeling experts are given a few specific questions to answer, apply the models almost in isolation, and then present the results to the decision-makers at the end of the study (Figure 1). The decision-makers have difficulty in understanding and trusting the results of the modeling effort. A process oriented approach encourages an iterative, explorative,

learning process between the user (the decision-maker) and the model, and would help involve more people in the decision loop (Figure 2).

One DSS architecture for environmental modeling has a collection of documentation, databases, and different models instead of a single complex model (Walker, 1982; Fedra and Loucks, 1985). The philosophy of this type of system is to encourage hands-on learning and the exploration, definition, and synthesis of alternatives, regardless of whether the user is a modeling specialist or nonspecialist. Chapra and Canale (1987) discuss how communication between engineers and nontechnical people can also be improved with new DSS type technologies such as data management, interactive computing, graphics, and simulation models.

Software Technologies for Process Oriented Modeling

Several new software technologies show promise for overcoming some of the difficulties associated with process oriented modeling. Three of these technologies were applied to build OASIS, a decision support system for ground-water contaminant problems:

1. **Graphical Interfaces** permit the user to send and receive more information in a more natural and intuitive fashion than with conventional text-based interfaces.
2. **Hypertext** is a new field of information management, wherein text, graphics, and data management tools are linked together in large computer documents.
3. **Object Oriented Programming** is a new programming approach that encourages a modular approach to the development of complex software.

Graphical Interfaces

The conventional method to communicate with microcomputers has been to exchange strings of text through keyboards, printers, and low-resolution displays. This approach has a low level of communication and has limited the way people use computers. Recently, however, new approaches to computer/human interfaces are producing a "revolution in the way many people think about computing" (Crecine, 1986).

Graphical interfaces are an improved way to transfer information between the computer and the user. Graphical interfaces have a high-resolution bit-mapped screen, a pointing device such as a mouse, and software that uses pull-down menus, windows, and graphical symbols ("icons") that represent files or other objects in the computer (Figure 3). These elements are used together to exploit the expressive power of pictures and symbols, and allow the user to "point and click" at these symbols to operate the computer. This simplifies routine microcomputer tasks such as file management and eliminates the need to type special commands needed to use the computer.

The operating system for the Apple Macintosh personal computer is a well-established graphical interface that has enhanced the interaction between the computer and the user. Several studies have suggested that a graphical interface is easier to learn and use than a conventional text-

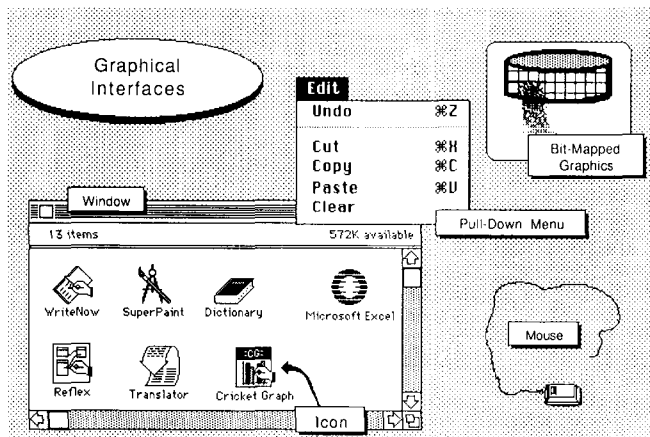


Fig. 3. Components of a graphical interface.

based interface (Peat, Marwick, Main & Co., 1987; Bulkeley, 1987). The development of the graphical Presentation Manager for IBM's PS/2-OS/2 computers and the current interest in a graphical interface for the UNIX operating system illustrate the importance the marketplace is placing on graphical interfaces for all types of microcomputers and workstations.

Several researchers have investigated the application of advanced interfaces to environmental modeling. Loucks, Taylor, and French (1985) describe the development of graphical computer displays for surface-water modeling and present several applications. A review of the recent literature pertaining to graphics in water resource problems is given by Loucks and Fedra (1987). Stephanopoulos (1987) describes the importance of the user interface and the problems in building interfaces for decision support systems for the chemical process field.

Hypertext

Hypertext describes a new type of information medium that is now appearing, where large amounts of diverse information are connected electronically in computers (Figure 4). We are very familiar with organizing, storing, and circulating knowledge using books, with the knowledge being presented sequentially, page after page.

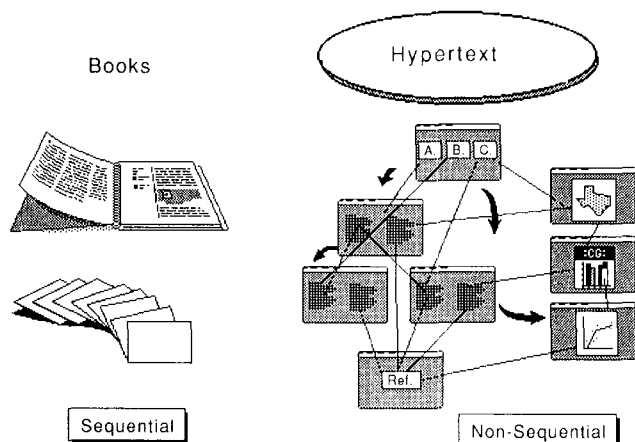


Fig. 4. Organization of knowledge in books vs. hypertext format.

With computers, information can be stored and accessed with a different architecture, one that encourages new associations, browsing, and easy navigation through large amounts of information (Nelson, 1987). Young (1986) describes hypertext this way:

“Like an encyclopedia, a Hypertext system connects ideas and references by directing readers to other documents. But, unlike an encyclopedia, it gives readers electronic bridges for reaching the referenced information quickly and easily. Readers can browse or create pathways through linked, cross-referenced, annotated documents having text, pictures, numerical data, computer graphics, music, or video with the click of a mouse button.”

On the surface, the development of hypertext documents appears to be nothing more than linking different screens of information with “buttons” that the user can “click” with the mouse to go to another screen. The construction of a polished hypertext system, however, requires careful design, implementation, and testing. Important hypertext issues include preventing user disorientation, converting linear paper-based documentation into hypertext, handling combinatorial explosions of information, and reducing the knowledge needed to use a hypertext system (Dede, 1988). A hypertext research community is now forming to address these issues and other problems related to developing large nonsequential electronic documents.

Object Oriented Programming

Some software experts feel that the most important consequence of artificial intelligence research will not be intelligent computer programs or knowledge-based systems; instead, artificial intelligence will lead to a new approach to computer programming (Waldrop, 1987). This new approach, object oriented programming, is a powerful programming paradigm that promises to change the way people develop models, interfaces, knowledge-based systems, and other software (Verity, 1987; Gevarter, 1987; Stephanopolous, 1987). Object oriented programming produces a remarkable change in point of view that increases the expressive power of a programmer (Kay, 1984).

The problems of conventional structured programming methods such as FORTRAN can be seen in the development of complex software where specifications and goals change as the program is being developed. Programming problems multiply when the program gets too large for one person to understand the overall data structure and the program logic (Verity, 1987). The conventional programming approach almost makes software development piecemeal operation, where it is difficult for more than a few people to work on the problem at once.

Artificial intelligence programming environments are characterized by a less structured, more exploratory approach, where the goals and the development process evolve together as the problem is understood and tested (Bobrow and Stefik, 1986). Artificial intelligence researchers developed highly interactive programming environments that simplify the building of large, rapidly changing programs. The result was a new approach to developing software called object oriented programming.

Object oriented programming is a programming style that creates a modular representation of knowledge, where each object is a piece of software that “knows” its properties (data) and methods (procedures), and responds to certain messages (commands). Familiar programming concepts such as data, procedures, and subroutines are replaced by “a community of semi-autonomous individuals—the objects” (Waldrop, 1987). Figure 5 illustrates three simple objects, each with their own data and procedures, reacting in three different ways to the message “invert.” A programmer working with these objects does not need to be concerned with how to invert an object; each object “knows” how to invert itself if the message to invert is given.

Some programmers suggest that the appeal of object oriented programming is not a particular technical advantage, but a more intangible quality that crosses a threshold of perception (Kaehler and Patterson, 1986). Programmers can tap directly into a familiar view of structuring knowledge based on things, concepts, and processes and avoid the effort of translating a system to fit the style of a conventional computer language. To put it simply, conventional structured programming asks the programmer to consider “What are my inputs and outputs?” and “How are my data transformed?” Object oriented programming asks “What are the things that I will be working with, and what are these things to do?” (Linowes, 1988).

One tangible benefit of object oriented programming is higher productivity for programmers. Some advocates claim that an object oriented approach is 25 times as productive as conventional systems (Verity, 1987) while others report improvements of four to six times (Waldrop, 1987). Object oriented programming is being used for a variety of problems on many different types of computers. Microcomputer object oriented languages include SMALLTALK, C++, Object Pascal, and, to some extent, the Macintosh HyperCard software.

Oasis: A Toolbox for Ground-Water Contaminant Modeling

The three technologies described above are now being used by the Rice University Department of Environmental Science and Engineering to develop a decision support system for ground-water contaminant modeling called OASIS. The research is funded by the Environmental Protection Agency’s Robert S. Kerr Environmental Research Laboratory. The goals of the project are to:

- Develop a computerized software toolbox for aquifer restoration problems with emphasis on contaminant transport modeling;
- Identify the important tools used by ground-water experts to restore contaminated aquifers;
- Develop documents and databases that the user can easily browse through to find and extract information;
- Rely on the three technologies described above to make interaction with the system intuitive and to make the system easier to build.

The first year of the two-year project has been spent in developing the 1700 computer screens of information

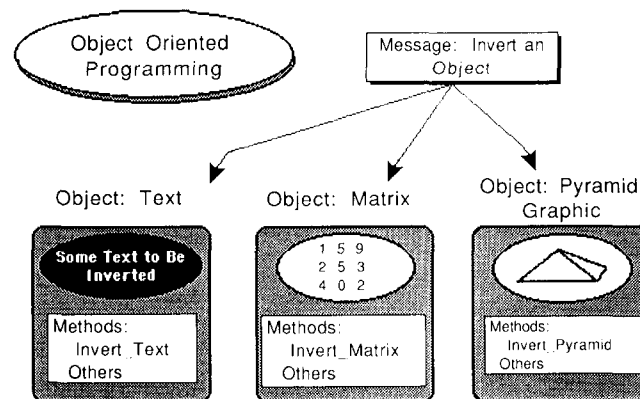


Fig. 5. Example of objects and messages in object oriented programming.

and installing the models that form OASIS. The second year will be spent conducting an evaluation program to debug and test the software.

OASIS was built using HyperCard, a software package supplied with the Apple Macintosh. In the HyperCard environment, each different screen of information is called a “card,” and cards are collected into groups called “stacks.” The cards within a stack are connected to each other by “links.” These links may take one of many forms: for example, a link may consist of a “button” which sends the user to a particular card, or a “field” which displays data taken from another card. In the same way, links may exist to cards in other stacks, or to other files and programs on the computer. HyperCard treats other programs and their data files as objects, with messages and links defining their role in the HyperCard environment.

The concept of independent objects in HyperCard extends beyond simple linking functions. HyperCard contains an object oriented programming language which allows developers to define more complex behaviors for an object. For example, a script (a small program for an object) could be written to respond to a user clicking the mouse by averaging a series of numbers, or computing the solution to an analytical equation. Despite the complexity of the behavior, however, an object remains an independent individual, completely defined by its responses to various messages. Many complex objects in the OASIS system were created as templates and distributed throughout the stacks without rewriting large chunks of specific code.

Buttons, fields, and links are all predefined objects in HyperCard. To create a button, no code need be written. A command found in a pull-down menu is selected, the new button appears and is moved and resized by mouse movements. Even the links needed to send the user to other cards are created using menu commands and mouse operations. These object oriented features of the HyperCard environment allowed scientists with relatively little programming experience to build parts of the system as plug-in “modules.” Eight environmental engineering students developed the current OASIS system in nine months with about 30-person months of effort. Only one member of the group was an experienced programmer, and most of the novice programmers began developing screens for the

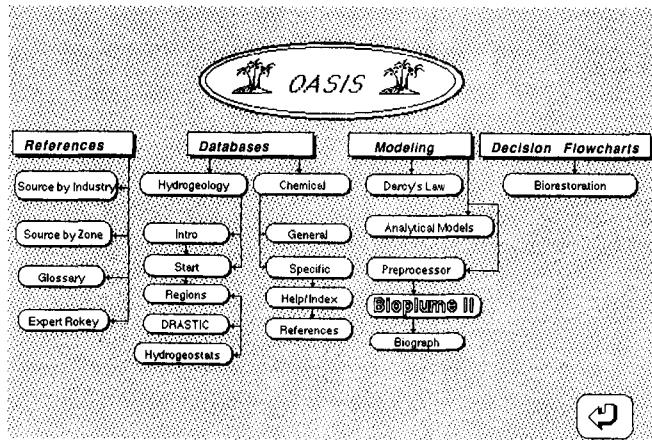


Fig. 6. Organization of OASIS stacks.

system after only a day of training. Most of the 1700 screens in OASIS were built without writing any lines of computer code; instead, the authors manipulated pre-existing software objects including cards, text fields, buttons, and pictures. Informal estimates by outside observers were that using conventional programming methods to develop OASIS would have taken at least five times as long.

The OASIS system currently occupies about 10 megabytes of hard disk storage. Figure 6 shows an outline of the system and the direct links between different stacks. Table 1 describes the relative size of each component of the system based on the number of cards in each stack. OASIS can be divided into three groups of HyperCard stacks: (1) reference libraries, (2) databases, (3) ground-water models and interfaces.

Reference Libraries

The reference libraries consist of text and figures that have been scanned into the computer from existing paper documents. Two of the 37 cards from a HyperCard stack describing sources of contamination are shown in Figures 7 and 8. Figure 7 shows an index to the different cards describing different types of sources while Figure 8 shows an information card for waste disposal wells. The libraries are designed as hypertext documents to allow the user to

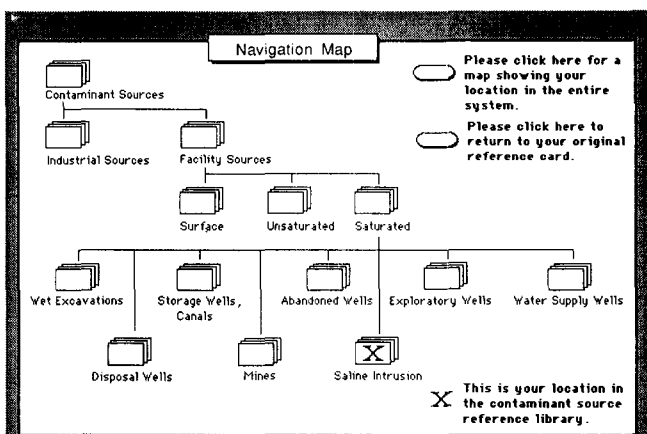


Fig. 7. Navigation map for source information stack.

Table 1. OASIS Software Toolbox – Description of Stacks

	No. of cards
I. Reference Library	
A. Hazardous waste by zone	37
B. Hazardous waste by industry	53
C. Literature review	143
D. Documentation	100
E. Glossary	14
II. Databases	
A. Chemical database	
1. General	167
2. Specific	450
a. Help/index	57
b. References	25
B. Hydrogeologic database	
1. Intro	35
2. Start	5
a. Regions	260
i. DRASTIC	10
Documentation	54
ii. Hydrogeostatistics	63
Documentation	19
III. Models	
A. Seepage velocity calculator	5
B. Analytical model	14
C. Bioplume II	
1. Preprocessor	20
a. Model help	100
b. Manual	20
2. Biograph (written in Pascal)	—
OASIS total system	1724

choose a path through the stack based on the specific information desired. This eliminates needless searching through large sections of the libraries not relevant to the current problem. Navigation is done using active buttons on each card, and is helped by a flowchart of the entire collection of cards (the stack) that shows the structure of the information in the library. This type of mapping facility has proven invaluable in preventing user disorientation, and allows very rapid movement from one section of the library to another.

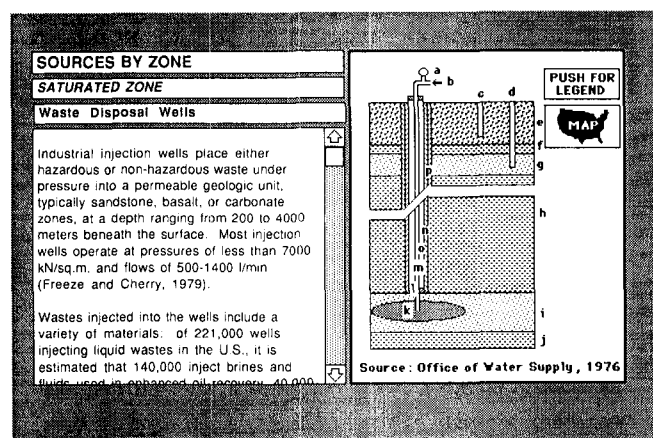


Fig. 8. Information card for waste disposal wells.

Chemical and Hydrogeologic Databases

Two chemical databases were incorporated into the system. The first is an existing expert systems chemical database (McClymont and Schwartz, 1987) which is being updated with new information on biodegradation. This stack is called the general chemical database and has 117 chemicals with one card of information per chemical. The specific chemical database contains a total of 582 cards divided into 18 common contaminants with 25 cards of

Physical data		Benzene	
Physical State	liquid		
Melting Point	5.5 - 6°C	2, 3, 6, 7, 206	
Boiling Point	80°C	1, 2, 3, 4, 206	
Specific Gravity	0.88	2, 3, 4, 5, 8, 206	
Vapor Pressure	75-76 mm Hg at 20°C	1, 14, 200, 206	
Vapor Density	2.7 - 2.8	6, 8, 10, 14, 200	
Solubility in Water	0.07 - 0.19	3, 5, 6, 7, 8, 10, 13, 206	
Solubility in Common Organic Solvents	alco, CS ₂ , eth, ac a, ace, CHCl ₃ , and CCl ₄	2, 4, 6, 7	
Fire Hazard Data			
Flash Point - Open Cup	21°F	14	
Closed Cup	12°F	3, 5, 6, 8, 10, 48, 206	
Autoignition Temperature	1000°F - 1076°F	3, 5, 6, 7, 8, 10, 13, 206	
Flammability Limits in Air (% by vol)			
Lower	1.3	8, 13, 14, 207	

Fig. 9. Data card from specific chemical database.

AB ⇌ A + B		Benzene	
<input type="checkbox"/> Combustion	<input type="checkbox"/> Oxidation/reduction	<input type="checkbox"/> Environmental Fate	
<input type="checkbox"/> Corrosion	<input type="checkbox"/> Photochemical	<input type="checkbox"/> In Natural Water	
<input type="checkbox"/> Explosion Hazards	<input type="checkbox"/> Polymerization	<input type="checkbox"/> Metabolism	
<input type="checkbox"/> Impurities	<input type="checkbox"/> Stability	<input type="checkbox"/> Reaction with air	
<input type="checkbox"/> Incompatibilities		<input type="checkbox"/> Reaction w/ water	

Fig. 10. Selection card from specific chemical database.

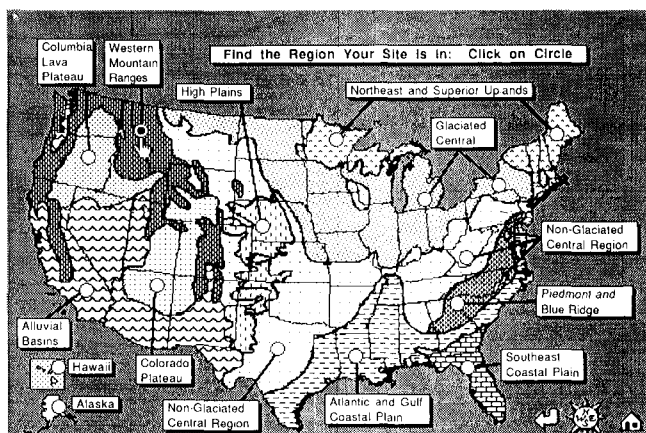


Fig. 11. Region map from hydrogeologic database.

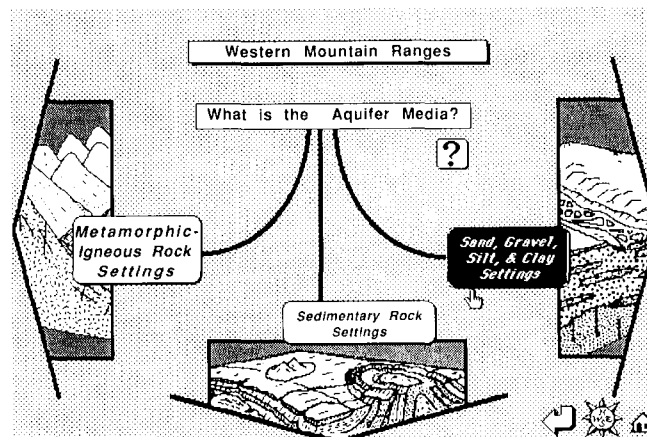


Fig. 12. Aquifer media card.

physical, chemical, toxicological, safety, and transport and fate parameters per chemical. A 350-page paper document (Ward *et al.*, 1981) was scanned into the Macintosh using an optical character reader and converted into hypertext format for easy access. The combination of the hypertext and search functions provided in HyperCard allow rapid and easy access to the data for a chemical in either the general or the specific database from a variety of search criteria. Two cards from the specific chemical database are shown in Figures 9 and 10.

A graphical hydrogeologic database was developed based upon 144 hydrogeologic "settings" from the NWWA's DRASTIC System (Aller *et al.*, 1987) and data obtained from a national survey of 400 hydrogeologic site investigations across the country (Newell *et al.*, 1988). The DRASTIC system is a combination of an aquifer classification system and an aquifer vulnerability index developed by a group of hydrogeologic experts. In OASIS, determination of the hydrogeologic setting of a site is a graphical procedure in which the user examines various graphical depictions of regions and settings and chooses the most appropriate one. Figures 11, 12, and 13 show some cards that would be seen if information about a coastal beach aquifer from Oregon was desired. The graphical interface of the OASIS system allows the user to see maps and block diagrams that define the hydrogeologic settings that form the structure of the database.

Fig. 13. Card for DRASTIC hydrogeologic setting.

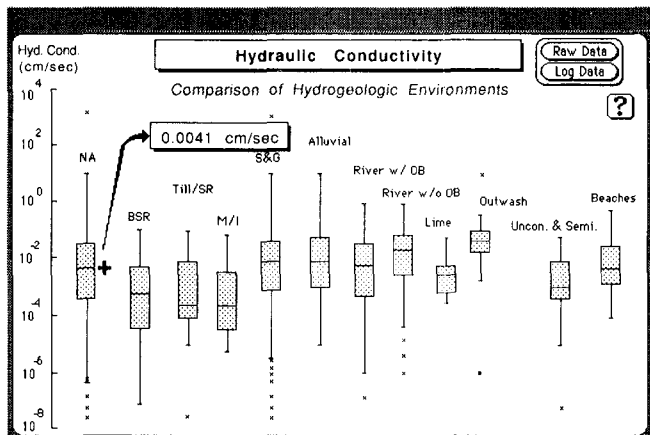


Fig. 14. Box plots from hydrogeologic statistics database.

```

Beaumont Modeling Contaminant Run 1 - Base Run. 7/12/88
5 1 19 1010800 1 7 0 100 2 9 1 0 1 0 0 0 0 0
1 00.001 0.3 10 0 0 0 0 50 50 0.1 0.5 1.0
11 6 -0.0006 100
10 7 -0.0006 100
0 6.8e-4
0 40.0
0 0.0
1 1.0
00000000000000000000
011111111111111111110
0100000000000000010
0100000010000000010
0100000000000000010
0100000000100000010
0100000000000000010
0100000000000000010
011111111111111111110
00000000000000000000
1 1.0 0.0 0.0
1 1.0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.46 4.6 4.6 4.6 4.6 4.6 4.6 4.5 4.4 4.4 4.3 4.3 4.2 4.1 4.0 3.9 3.8 0.
0.46 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3.8 0.
0.46 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3.7 0.
0.46 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3.7 0.
0 4.5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2.7 0

```

Fig. 15. Sample conventional input data for BIOPLUME II.

Once a setting has been selected, the user can choose from two options. A DRASTIC vulnerability index can be calculated for different sites, with a higher DRASTIC index meaning the aquifer is more vulnerable to contamination. The second option is to enter the hydrogeologic database. Figure 14 shows one part of the database: box plots of hydraulic conductivity data collected from the survey of 400 hydrogeologic site investigations across the country. The box plots show the distribution of the data, such as the outliers, 25th and 75th percentile of the data, and the median value of the data. This data can be used for statistical modeling or for showing the variability in ground-water data.

Ground-Water Models

The system contains a collection of ground-water models: a seepage velocity calculator, two analytical solute transport models, and a numerical biodegradation model. This provides a spectrum of models that the user can apply to different types of problems. The analytical models and the seepage velocity calculator are written in a combination of Hypertalk, the HyperCard programming language, and Pascal.

The upper level model in the system is Bioplume II, developed at Rice University by Rifai *et al.* (1988), which allows the prediction of naturally occurring biodegradation and in-situ bioremediation at waste sites. The model is based

upon the USGS solute transport code by Konikow and Bredehoeft (1978) and work by Borden and Bedient (1986), and simulates the transport and reaction of oxygen and a biodegradable solute. Bioplume II is written in Fortran and was ported to the Macintosh from an existing MS-DOS version of the model with two days of effort by an experienced Macintosh Fortran user. Access to the Bioplume II code from the OASIS system follows the object oriented paradigm. A message is sent to HyperCard from OASIS to execute the model. This message is all the OASIS system needs to "know" about Bioplume II. The execution of the program is handled by HyperCard and the Macintosh operating system. Additional information (input data) is transferred to the model via text files stored on the computer's hard disk. Knowledge about the input format required by the program is stored in the preprocessor. Similarly, the knowledge about reading output files is stored in the postprocessor. Thus, the model and the pre- and postprocessors form a modular package which can be reproduced easily for other models and inserted into the OASIS system.

The graphical data preprocessor was developed for Bioplume II using HyperCard to make the manipulation of input data more intuitive and less confusing. The conventional method used for data input involves typing into a text file a combination of lines and arrays of data that must be entered in a specific, sequential order with no on-screen indication of what the numbers represent (Figure 15). The graphical preprocessor removes this sequential limitation, and provides on-screen information regarding the purpose and meaning of the data. It also allows the user to access databases and help screens in order to apply as much knowledge as possible to the estimation of a parameter value. To help in the creation of array-style data, painting tools were developed to allow the user to "draw" the data array on-screen as patterns in a grid, and later convert those patterns to specific numbers. Figure 16 shows this graphical approach applied to entering a hydraulic conductivity array. Graphical entry of injection and pumping well locations are shown in Figure 17.

Figure 18 shows a help card for the dispersivity parameter based on work done for the expert ROKEY

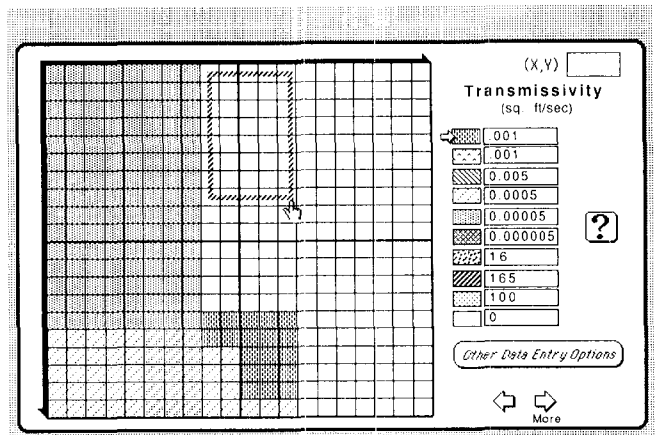


Fig. 16. Graphical preprocessor card for entering hydraulic conductivity data.

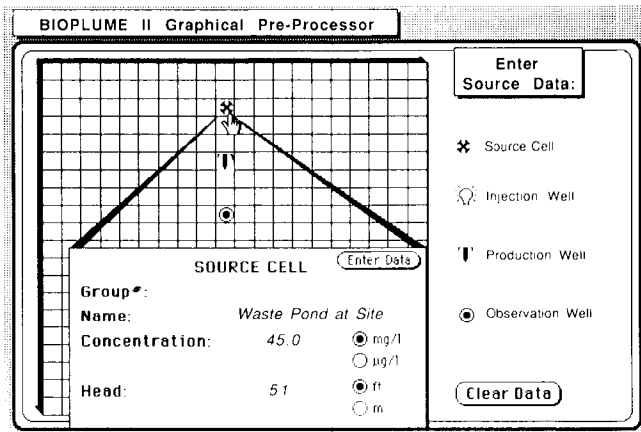


Fig. 17. Graphical preprocessor card for entering well locations.

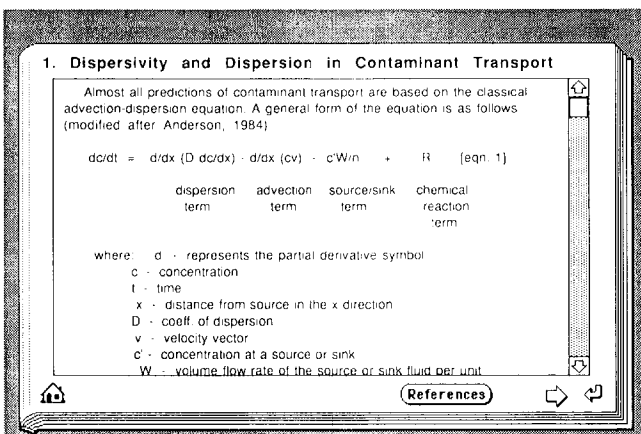


Fig. 18. Example of help card for dispersivity.

system (McClymont and Schwartz, 1987). Text files from the ROKEY system were ported from an MS-DOS system to the Macintosh, divided into help cards, and linked to the preprocessor. The user can access 100 different help cards by clicking on help buttons within the preprocessor. Other help cards include units conversion calculators, empirical computations of parameter values, and literature values for selected parameters.

A graphical postprocessor, BioGraph, was designed to make model calibration and data analysis much faster and more intuitive. BioGraph was written completely in Pascal to improve performance. The entire program is menu- and mouse-driven. Plume behavior in space and time appears on a grid with grey-scale shading to represent concentrations of contaminant or oxygen. Mouse clicks in the model grid invoke a pop-up window showing temporal, longitudinal, or transverse concentration profiles (Figures 19 and 20). The preprocessor and BioGraph are designed to make the manipulation and analysis of modeling data less tedious to the user. Instead of working with large arrays of numbers for input and output, the user can have the system automatically display the conditions at a site graphically.

Discussion

Although the OASIS system has a large collection of tools for ground-water contamination problems, it is difficult to get a sense of how these tools are used together to solve a specific problem. To illustrate the potential advantages of a software toolbox, the OASIS approach to ground-water contamination modeling is compared against the conventional approach. In the example below, a modeler uses the BIOPLUME II model at a hydrogeologic site investigation.

A modeler often starts by collecting chemical and hydrogeologic data from several external databases. Some data are used directly in the model while other data are used as background information. The conventional approach has the modeler paging through reports, books, and databases to find the data. For the experienced modeler, this process may be relatively painless, as an expert usually has a comprehensive personal library that has been collected over the years. The beginning modeler, however, is faced with starting a library of external data from scratch, greatly increasing the difficulty and time needed to build the model.

The existing OASIS databases provide a built-in library of contaminant data and hydrogeologic settings. The information does not cover all possible problems, but provides several megabytes of data accessible by pointing and clicking the mouse. More importantly, a hydrogeologist can add data, entire databases, and models to the existing

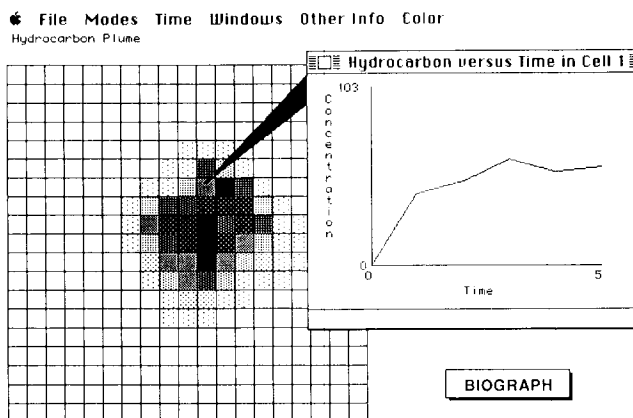


Fig. 19. BIOGRAPH postprocessor showing concentration vs. time at point in model.

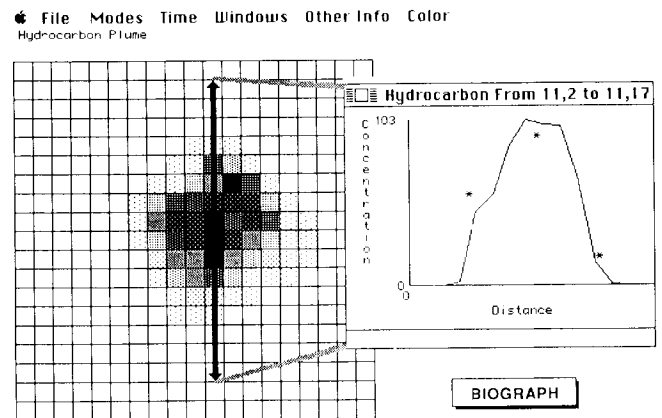


Fig. 20. BIOGRAPH postprocessor showing concentration vs. distance in model.

OASIS system in the same way he or she adds to a personal library of hydrogeologic and contaminant information. Scanning technology and commercial ground-water software are becoming more common, and the HyperCard system is open-ended and easily modified by each user (over two-thirds of all current HyperCard users have modified their systems to some extent). The advantage to OASIS is that the system provides a baseline set of databases and an architecture for starting a personal electronic library.

BIOPLUME II requires extensive field data from a site and external data from databases. Data from monitoring wells and other sources in the field are used to build maps of hydraulic conductivity, saturated thickness, and provide data on other parameters. The conventional approach forces the modeler to type in arrays of numbers in fields that represent mapped parameters. The model is run and the model output is graphed by hand or by an external graphing package. In summary, the modeler must spend time translating maps to input data and then translating the output numbers back to maps and graphs.

OASIS provides advantages to both experienced modelers and beginning modelers for the manipulation of data. The preprocessor and the graphical postprocessor, BIOGRAPH, reduce time devoted to translating numbers. The system does some of the mundane and repetitive data management tasks that tend to slow down experienced modelers. The beginning modeler also can take advantage of these graphical tools, and has the additional support of an extensive set of help screens.

Conclusions

Decision support systems are a class of software that can help make ground-water modeling technology more accessible and useful to ground-water scientists and engineers. A decision support system is a collection of software tools (such as models, documents, databases, and data management software) that are integrated together with an easy-to-use interface. The decision support system is well-suited for solving types of problems that do not have a clearly defined structure, so that conventional structured programming is difficult to apply. Expert systems and decision support systems both deal with poorly structured problems; however, decision support systems are designed for broad, nonroutine problems where the user controls the problem-solving approach.

Three software technologies were used to develop OASIS, a decision support system for ground-water contaminant transport modeling. Graphical interfaces make the system easier to learn and use compared to text-based interfaces. Hypertext concepts deal with the development of complex electronic information systems, and were applied to the architecture of the OASIS system. Finally, object oriented programming permitted scientists with little programming experience to develop OASIS by manipulating preexisting software objects instead of writing computer code, greatly increasing the productivity of the project team.

The philosophy behind a decision support system such as OASIS is to encourage hands-on learning and a per-

sonal, explorative approach to solving ground-water problems by using a combination of "tools" in the system. It contains computerized documentation, two chemical databases with 117 different chemicals, a hydrogeologic database derived from a survey of 400 sites across the country, and BIOPLUME II, a numerical 2-D solute transport and biodegradation model. OASIS has over 1700 different computer screens and runs using HyperCard software on a Macintosh personal computer. The system now requires 10 megabytes of storage on a hard disk; storage was not considered to be a constraint because of the rapidly falling cost of conventional hard disks and the advent of new mass storage technologies.

OASIS is designed to be an open-ended software system where new tools can be added by other groups or by individual users. Other future enhancements might include more knowledge-based decision trees, links to larger databases, and a wider variety of models taken from the large library of existing MS-DOS programs. These enhancements and personal modifications by the user are easier to incorporate into OASIS than a system written using conventional languages because of the object oriented nature of HyperCard and the graphical interface of the system.

One potential drawback to a system such as OASIS is that the modeling technology may be misused by people who do not have the necessary background to use the models correctly. Models are misused today, and by making the technology more accessible the potential for harm may increase. One mitigating factor is that a well-designed decision support system encourages the learning process and experimenting with the data and the model features. Knowledge of how a model responds under different conditions is good insurance against misuse.

Acknowledgments

We would like to thank the U.S. EPA and the Robert S. Kerr Environmental Research Lab for funding most of this research, and acknowledge the contributions of Joe Williams and James McNabb of the U.S. EPA. We also would like to thank the American Petroleum Institute for funding the hydrogeologic database and the National Water Well Association for technical support. We would like to acknowledge the contributions of Tony Gorry of the Baylor College of Medicine, the coprincipal investigator of the project; the panel of experts: Frank Schwartz, Randy Charbeneau, and Dick Raymond; and the OASIS project team: Julie Smythe, Loren Hopkins, David Todd, Lois Ongley, Sarah Alder-Schaller, Jessica Gould, Peter Nevin, Kathleen Debo, and Melissa Schilhab.

References

- Aller, L., T. Bennett, J. Lehr, R. J. Petty, and G. Hackett. 1987. DRASTIC: A standardized system for evaluating ground water pollution potential using hydrogeologic settings. U.S. EPA. EPA-600/2-87-035.
- Bobrow, D. G. and M. J. Stefik. 1986. Perspectives on artificial intelligence programming. *Science*. v. 231, pp. 951-957.
- Borden, R. and P. B. Bedient. 1986. Transport of dissolved hydrocarbons influenced by oxygen limited bio-

- degradation, 1. Theoretical development. *Water Resources Research*. v. 22, no. 13, pp. 1973-1982.
- Bulkeley, W. M. 1987. Uncovering the hidden costs. *Wall Street Journal*. June 12, p. 14D.
- Chapra, S. C. and R. P. Canale. 1987. Personal computers and environmental engineering. *Environmental Science and Technology*. v. 21, no. 9, pp. 832-837.
- Crecine, J. P. 1986. The next generation of personal computers. *Science*. v. 231, pp. 935-943.
- Dede, C. 1988. The four great problems of HyperMedia and their solutions. Hypermedia '88 Conference, University of Houston-Clear Lake, TX. September 14-15, 1988.
- Domenico, P. A. 1972. *Concepts and Models in Ground-water Hydrology*. McGraw Hill, New York.
- Doukidis, G. I. 1988. Decision support system concepts in expert systems: an empirical study. *Decision Support Systems*. v. 4, pp. 345-354.
- Fedra, K. and D. P. Loucks. 1985. Interactive computer technology for planning and policy modeling. *Water Resources Research*. v. 21, no. 2, pp. 114-122.
- Feigenbaum, E. A. 1977. The art of artificial intelligence: Themes and case studies of knowledge engineering. *Int. Joint Conf. on Art. Intell.* v. 5, pp. 1014-1029.
- Gevarter, W. B. 1987. Introduction to artificial intelligence. *Chemical Engineering Progress*. v. 83, no. 9, pp. 21-37.
- Gorry, G. A. and M. S. Scott-Morton. 1971. A framework for management information systems. *Sloan Management Review*. pp. 55-70.
- Kaehler, T. and D. Patterson. 1986. A small taste of small-talk. *Byte*. v. 11, no. 8, pp. 145-159.
- Keen, P. 1980. Decision support systems: A research perspective. *Decision Support Systems: Issues and Challenges*. IIASA Proceedings Series, G. Fick and R. H. Sprague, eds. Pergamon Press, Oxford.
- Konikow, L. F. and J. D. Bredehoeft. 1978. Computer model of two-dimensional solute transport and dispersion in ground water. *U.S. Geological Survey Techniques of Water Resources Investigations*. Book 7.
- Kay, A. 1984. Computer software. *Scientific American and W. H. Freeman and Co.*, New York. Originally in *Scientific American*. Feb. 1984.
- Lara, F. and W. M. Sachs. 1978. Modeling of the environment through a system of models. *Applied Math. Models*. v. 2, pp. 216-218.
- Linowes, J. S. 1988. It's an attitude. *Byte*. v. 13, no. 8, pp. 219-224.
- Loucks, D. P. and K. Fedra. 1987. Impact of changing computer technology on hydrologic and water resource modeling. *Reviews of Geophysics, American Geophysical Union*. v. 25, no. 2, pp. 107-112.
- Loucks, D. P., J. Kindler, and K. Fedra. 1985. Interactive water resources modeling and model use: an overview. *Water Resources Research*. v. 21, no. 2, pp. 95-102.
- Loucks, D. P., M. R. Taylor, and P. N. French. 1985. Interactive data management for resource planning and analysis. *Water Resources Research*. v. 21, no. 2, pp. 131-142.
- Ludvigsen, P. J., W. J. Grenney, D. Dyreson, and J. M. Ferrara. 1986. Expert system tools for civil engineering applications. *Computing in Civil Engineering*. TCCP Div/ASCE, Boston, MA. April 8, 1986, pp. 18-29.
- McClymont, G. L. and F. W. Schwartz. 1987. Development and application of an expert system in contaminant hydrogeology: the expert ROKEY system. *National Hydrology Research Institute*. DSS File No.: 52SS.KN107-5-4334, Serial No.: OSS85-00149.
- McKinney, K. C. and D. P. Loucks. 1986. Interactive ground water quality modeling system. NTIS NSF/Eng-86040.
- Nelson, T. H. 1987. *Literary machines*, edition 87.1. Project Xanadu. San Antonio, TX.
- Newell, C. J., L. P. Hopkins, and P. B. Bedient. 1988. Hydrogeologic database for ground water modeling. Report for the American Petroleum Institute, Washington, DC.
- Peat, Marwick, Main & Co. 1987. *Macintosh benefits survey*. Peat, Marwick, Main & Co.
- Rifai, H. S., P. B. Bedient, J. T. Wilson, K. M. Miller, and J. M. Armstrong. 1988. Biodegradation modeling at a jet fuel spill site. *J. Environmental Engineering Div., ASCE*. v. 114, no. 5, pp. 1007-1028.
- Sheil, B. 1983. Power tools for programmers. *Datamation*. Feb., pp. 131-144.
- Sprague, R. H. and E. Carlson. 1982. *Building Effective Decision Support Systems*. Prentice Hall.
- Sprague, R. H. 1987. DSS in context. *Decision Support Systems*. v. 3, pp. 197-202.
- Stephanopoulos, G. 1987. The future of expert systems in chemical engineering. *Chemical Engineering Progress*. v. 83, no. 9, pp. 44-51.
- Verity, J. W. 1987. The OOPS revolution. *Datamation*. May, p. 73.
- Waldrop, M. M. 1987. Artificial intelligence moves into the mainstream. *Science*. v. 237, pp. 484-485.
- Walker, W. E. 1982. Models in the policy process: past, present, and future. *Interfaces*. v. 12, no. 5, pp. 91-100.
- Ward, C. H., R. C. Allison, and S. M. Pier, editors. 1981. *Hazardous Substance Resource Book, Vol. I-Vol. V*. USEPA Emergency Response Division, Office of Hazardous Emergency Response. WH-548-B, June.
- Young, J. S. 1986. *Hypermedia*. *MacWorld*. March, pp. 116-121.

* * * * *

Charles J. Newell has a Ph.D. in Environmental Engineering from Rice University. He graduated with a B.S. in Chemical Engineering in 1978 and an M.S. in Environmental Engineering in 1980 from Rice University and spent five years working as an environmental consultant before beginning his Ph.D. program in 1985. Newell's research interests are in computer modeling, hydrogeologic databases, and the application of artificial intelligence and decision support software to environmental problems.

John F. Haasbeek received a B.S. in Civil Engineering from Rice University in 1987, where he is currently completing the requirements for an M.S. in Environmental Engineering. His research focuses on the application of modeling technology to the design of in-situ aquifer remediation systems with specific emphasis on a knowledge-based approach to parameter estimation.

Philip B. Bedient is a Professor of Environmental Engineering at Rice University. He holds a Ph.D. in Environmental Engineering from the University of Florida, Gainesville, in 1975. Bedient is a researcher for the Center for Ground Water Research and directed the development and field application of the BIOPLUME model. His recent research interest is in the development of decision support systems.